2. Amelia

Program Name: Amelia.java **Input File: amelia.dat**

Amelia is now a TA for ECS 1100 at her college program. She wants your assistance creating a sorting program to determine who the best students are, so that we can find out who deserves extra credit. The sorting algorithm is outlined as follows:

- First, sort the students based on attendance percentage (classes attended/total classes), as the students who go to all the classes deserve the best treatment (the professor's words, not ours), with the higher attendance percentage coming first.
- Next, if attendance is the same between two students, sort by improvement average between exam 1 to exam 2, and exam 2 to exam 3 (((exam2 – exam1) + (exam3 – exam2))/2), with the higher improvement coming first.
- Next, if the improvement scores are the same between two students, sort by number of daily quizzes completed, with the most quizzes coming first.
- If all other requirements are the same, sort alphabetically by last name then first name.

Input: The input will begin with an integer, `num` (`0 < num <= 1000`), denoting the number of test cases to follow. Each test case will begin with one integer, `n` (`0 < n < 1000`), denoting the number of students in the class to be sorted. Each line will begin with two strings, separated by spaces, denoting the first and last names of the student in question, respectively. These strings will be followed by 6 integers, ca, tc, e1, e2, e3, and dq, denoting the classes attended, total classes, exam 1 score, exam 2 score, exam 3 score, and daily quizzes completed, respectively. Each of the student records will appear on its own line, and all values (string and integer both) will be separated by spaces.

Output: Output the names of the students, first name then last name separated by spaces, followed by a space, followed by the average of the 3 exam grades rounded to 2 decimal places, each on its own line, in sorted order. Every test case should be followed by a line of 10 asterisks.

Sample input:

```
2
3
Benjamin Armstrong 10 11 82 89 95 9
Derrick Martin 8 11 70 80 90 11
Matthew Sheldon 10 11 92 93 94 11
2
William Armstrong 8 9 80 90 100 7
AppleBottom Jeans 6 9 70 85 100 8
```

Sample output:

```
Benjamin Armstrong 88.67
Matthew Sheldon 93.00
Derrick Martin 80.00
**********
William Armstrong 90.00
AppleBottom Jeans 85.00
**********
```